

Treball Fi de Màster  
**Màster en Enginyeria Informàtica**

# **Apoderament d'una xarxa d'actuadors amb integració en un gestor web**

## **Memòria**

**Autor:** Marc Borràs Conte  
**Director:** Manel Frigola  
**Convocatòria:** Octubre 2017



**Facultat d'Informàtica de Barcelona**





## Resum

Aquest projecte té per objectiu oferir una forma fàcil i ràpida de crear una xarxa de sensors i actuadors. Es pretén facilitar la tasca d'incorporar un nombre elevat de dispositius a un entorn de control via web.

Per crear i controlar la xarxa, s'ha desenvolupat una aplicació per a smartphones que utilitza NFC per identificar un dispositiu. Gràcies a aquesta tecnologia, quan l'usuari apropa un smartphone, l'aplicació detecta un tag NFC, llegeix el seu identificador i inicia una connexió Bluetooth. Un cop afegit a la xarxa, l'usuari pot controlar cada dispositiu de forma independent des de la mateixa aplicació.

Com que un dels àmbits on Internet of things és més utilitzat és en la domòtica, es va decidir donar als usuaris la possibilitat d'integrar la xarxa construïda amb una plataforma d'automatització per a smarthomes anomenada Home Assistant. Aquestes plataformes estan pensades per facilitar el control dels sensors i actuadors que formen una casa intel·ligent.

Per compartir la xarxa des de l'aplicació per a smartphones a Home Assistant, es va dissenyar un software amb python que actua com a servidor i està a l'espera de connexions HTTP. A més de rebre la xarxa, aquest software també s'encarrega de fer de pont entre Home Assistant i els dispositius Bluetooth cada cop que l'usuari vol controlar-ne un.

Per comprovar el funcionament del sistema, es van desenvolupar uns prototips amb Arduino i un mòdul Bluetooth. Aquests prototips simulen dispositius comercials que implementen el protocol de control definit en el projecte i es poden controlar tant des d'un smartphone com via web gràcies a Home Assistant.

Finalment, es va fer servir un dels prototips i el software desenvolupat per construir un rec intel·ligent. Es va definir una regla d'automatització que per a valors d'humitat massa baixos, activa una bomba d'aigua per regar una planta i a més envia una notificació per telegram.

## Sumari

Glossari .....	1
1.Introducció.....	2
1.1. Estructura de la memòria .....	3
2. Objectius .....	4
2.1. Abast del treball .....	4
2.2.Especificacions .....	5
3. Estudi d'alternatives .....	6
3.1. Tecnologies de comunicació .....	6
3.1.1 Bluetooth LE .....	7
3.1.2. Wi-Fi .....	8
3.1.3. ZigBee .....	9
3.1.3. Z-Wave .....	10
3.1.4. NFC .....	10
3.1.5. Elecció Final .....	11
3.2. Plataforma d'automatització .....	12
3.2.1. Home Assistant.....	13
3.2.2. OpenHab .....	14
3.2.3. Comparativa .....	15
4. Disseny .....	16
4.1. Agents del sistema.....	16
4.1.1. Network Manager .....	16
4.1.2. Home Nexus .....	18
4.2. Dispositius prototip.....	20
4.2.1. Components del dispositiu .....	20
4.2.2. Estats del dispositiu .....	21
5. Implementació .....	22
5.1. Home Nexus .....	22
5.1.1. Home Assistant Bridge .....	22

5.1.2. Network Manager Bridge .....	25
5.1.3. Device Manager.....	25
5.1.4. Data Manager .....	26
5.2. Network Manager.....	26
5.2.1 Adició de dispositius .....	26
5.2.2. Control de dispositius .....	28
5.2.3. Interacció amb Home Assistant .....	29
5.3. Interacció entre dispositius.....	30
6. Integració i validació del sistema.....	31
6.1. Dispositius Prototip .....	31
6.1.1. Hardware .....	31
6.1.2. Software.....	34
6.1.3. Muntatge.....	36
6.2. Exemple d'ús: Rec automàtic amb notificació.....	37
7. Planificació .....	39
8. Pressupost .....	40
8.1. Hardware dels prototips .....	40
8.2. Personal.....	40
8.3. Equipament.....	41
8.4. Cost total.....	41
9. Conclusions.....	42
10. Bibliografia .....	43

## Glossari

**Bluetooth:** Tecnologia de comunicació sense fils que utilitza la radiofreqüència per transmetre informació entre dispositius.

**BLE (Bluetooth Low Energy):** Mode del protocol Bluetooth introduït en la versió 4.0 que redueix el consum dràsticament en connexions breus i puntuals.

**NFC (Near field communication):** Tecnologia utilitzada per comunicar dades sense fils a poca distància.

**IOT (Internet Of Things):** Connexió entre objectes d'ús quotidià entre si per tal d'intercanviar informació i actuar en conseqüència.

**HTTP (Hypertext Transfer Protocol):** Conjunt de regles que permeten la transferència de fitxers en diferents formats a través d'internet.

**MQTT (MQTT):** Protocol per transmetre informació, utilitzat sobre TCP i que es basa en una estructura client - subscriptor.

**HA (Home Assistant):** Plataforma d'automatització que permet controlar, recollir dades i crear interaccions entre els diferents dispositius que formen una casa intel·ligent.

**Yaml:** Format de serialització de dades dissenyat per tal que sigui fàcil llegir-lo per humans.

# 1. Introducció

En els últims anys una de les novetats que ha guanyat més força en el món de la tecnologia és Internet of Things (IOT). Des de bombetes fins a recs automàtics, cada cop s'ofereixen més dispositius capaços de ser controlats des d'un smartphone o amb un navegador d'internet.

Un dels àmbits on IOT està prenent més força és a la domòtica. Gràcies a aquestes tecnologies es poden monitoritzar i automatitzar molts processos que poden facilitar la vida quotidiana així com millorar la seguretat dels habitatges. Recentment han aparegut plataformes que unifiquen el control de dispositius des d'un mateix lloc i ofereixen possibilitats d'afegir interaccions entre ells.

A mesura que augmenta el nombre d'aquests dispositius IOT, apareix el problema de com identificar de forma fàcil i eficient quins pertanyen a cada usuari per tal que cadascú pugui controlar tan sols els seus. Per exemple en el cas d'una bombeta intel·ligent, si diverses persones d'un mateix bloc compren el mateix model, alhora d'encendre-la caldrà haver-la diferenciat de la resta abans de permetre'n el control. En aquest treball es cercarà una forma amb la qual els fabricants de dispositius IOT, puguin facilitar a un usuari sense gaires coneixements d'informàtica la creació d'una xarxa amb els seus productes. Es vol que aquest mètode sigui el més genèric possible per tal que funcioni amb una àmplia gamma de dispositius.

Aquestes plataformes d'automatització, es facilita en gran mesura la creació de tasques automàtiques i obren un ampli ventall de possibilitats. Per aquest motiu se li ha donat molta importància a que la xarxa creada per l'usuari es pugui integrar amb una d'aquestes plataformes. A causa de que els sensors són necessaris per crear aquestes automatitzacions, a més de permetre encendre i apagar els actuadors, s'oferirà la possibilitat de llegir sensors.

## 1.1. Estructura de la memòria

En el moment de redactar la memòria, s'ha decidit dividir-la en nou capítols:

1. **Introducció:** Es fa una primera introducció al lector explicant el motiu d'aquest treball.
2. **Objectius i especificacions:** Es mostren els diferents objectius que es pretenen assolir i quines especificacions haurà de complir el treball un cop finalitzat.
3. **Estudi d'alternatives:** En aquest capítol es veuran les tecnologies actuals que es podrien utilitzar en el projecte i es seleccionaran les que s'adaptin millor al projecte.
4. **Disseny:** La funció d'aquest apartat és definir quins agents intervindran al sistema, com s'estructuraran i quina relació tindran entre ells.
5. **Implementació:** Partint del disseny definit en el capítol anterior, es desenvoluparà el software necessari per tal que es compleixin les especificacions del projecte.
6. **Validació i integració del sistema:** Aquí es parlarà de com s'ha construït el dispositiu que s'utilitzarà per provar que els diferents agents que intervenen en el sistema interactuïn bé entre ells.
7. **Planificació:** Aquest capítol descriurà com s'ha planificat el desenvolupament de cada bloc del treball, mostrant quantes hores s'ha dedicat a cada part i com s'han distribuït al llarg del temps.
8. **Pressupost:** En aquest apartat es mostra el cost dels recursos utilitzats, tant materials com personals i es fa una estimació del cost real que tindria el projecte en cas que s'hagués desenvolupat dins d'una empresa.
9. **Conclusions:** Per finalitzar la memòria, es resumeixen les conclusions a les quals s'ha arribat després de desenvolupar tot el projecte.



## **2. Objectius**

El principal objectiu d'aquest treball és oferir a fabricants actuadors i a usuaris sense coneixements d'informàtica una forma fàcil per crear i gestionar una xarxa d'aquests dispositius des d'un smartphone.

A més com a objectiu addicional s'ha decidit oferir a usuaris més avançats la possibilitat d'integrar aquesta xarxa a una plataforma d'automatització per a smarthomes. Aquestes plataformes permeten afegir dispositius compatibles com televisors, alarmes, termòstats i molts més amb el fi de poder controlar-los via web.

### **2.1. Abast del treball**

En el transcurs d'aquest treball es tractaran els següents punts:

- Disseny d'una aplicació per a smarthomes capaç de crear / controlar una xarxa de d'actuadors amb una estructura mestre / esclau.
- Integració d'aquesta xarxa amb una plataforma d'automatització per a smarthomes. per tal oferir la possibilitat de controlar-la via web.
- Construcció d'un prototip per comprovar el funcionament del sistema.

Per contra, degut a la limitació de la durada del projecte no es podrà tractar els següents punts:

- Suport de múltiples sistemes operatius.
- Suport de múltiples plataformes d'automatització.
- Un entorn gràfic per a automatització de tasques.

## 2.2. Especificacions

A continuació es mostra una llista amb les operacions que es podrà fer amb el resultat del projecte:

- Crear una xarxa de dispositius compatibles fent ús de NFC.
- Transferir una xarxa creada amb un smarthome a la plataforma d'automatització.
- Encendre i apagar un actuador compatible des d'un smartphone.
- Encendre i apagar un actuador compatible des de la plataforma d'automatització per a smarthomes.
- Llegir les dades d'un sensor compatible des d'un smartphone.
- Llegir les dades d'un sensor compatible des d'un smartphone.
- Afegir / eliminar dispositius de la xarxa.

### **3. Estudi d'alternatives**

En aquesta primera etapa del projecte, es veuran les diferents tecnologies que es podrien utilitzar per complir els objectius del projecte, així com les plataformes d'automatització que més s'utilitzen actualment. S'investigarà quines s'adaptin millor a les especificacions definides en l'apartat anterior i es farà una selecció de quines s'utilitzarà.

#### **3.1. Tecnologies de comunicació**

Amb l'objectiu de seleccionar la tecnologia de comunicació més adient, s'ha fet una investigació de les més usades en el camp d'internet of things. S'ha determinat que les característiques més importants que haurà de tenir aquesta tecnologia són les següents:

- Baix consum: Molts dispositius d'internet of things estan dissenyats per alimentar-se d'una bateria. Com que serà necessari que estiguin disponibles en tot moment, cal que la tecnologia seleccionada tingui un consum reduït. En concret, cal que sobretot hi hagi un consum baix quan els dispositius es trobin en espera. És així perquè transmissió de dades tan sols ocuparà una mínima part del temps, mentre que quasi tota l'estona el dispositiu estarà a l'espera de ser connectat.
- Compatibilitat amb smartphones: Es desitja que l'usuari sigui capaç de controlar els dispositius des d'un smartphone, per tant cal que aquesta tecnologia hi sigui compatible.
- Velocitat de connexió: Quan l'usuari premi un botó per encendre un actuator, es vol observar la resposta el més ràpid possible.

Per contra, no serà molt important:

- Velocitat de transmissió: Degut a que les comunicacions seran molt breus (de l'ordre de bytes) i puntuals, no es requereix d'una velocitat de transmissió gaire elevada.
- Gran abast: Els dispositius per als quals està pensat aquest projecte són per ser controlats a mitjana distància, com a molt d'habitació a habitació.

### 3.1.1. Bluetooth LE

El Bluetooth és una tecnologia de comunicació per radiofreqüència definit per l'empresa sueca Ericsson Mobile l'any 1989 i que es va desenvolupar per fer auriculars sense fil. Actualment és una de les tecnologies més utilitzada arreu del món i està disponible per la majoria de dispositius d'ús diari com smartphones, tablets o ordinadors.

L'any 2010 es va llançar la versió 4.0 del protocol que incorporava un mode de baix consum anomenat Bluetooth Low Energy (BLE). Aquest mode es va implementar per fer possible que dispositius que intercanvien poca informació de forma puntual poguessin fer ús de Bluetooth sense que aquest tingués un gran impacte en la seva autonomia.

La diferència de consum entre la versió clàssica de Bluetooth i la Low Energy, ve donada perquè el LE està en mode repòs la major part del temps i tan sols es desperta quan s'inicia una connexió. Per aquest motiu, per aplicacions amb connexions curtes i puntuals serà molt més eficient que la versió clàssica de Bluetooth.

A més de millorar el consum respecte la versió clàssica, si es compara aquest amb altres tecnologies, també es veu que és molt reduït: quan està a l'espera de ser connectat, el consum és menor a 1mA i quan està transmetent informació pot arribar als 30 mA.

Els dispositius BLE es relacionen entre ells seguint una estratègia client servidor. Per exemple, un sensor Bluetooth actuaria com un servidor al qual el client (dispositiu que vol llegir-lo) s'hi connecta quan necessita informació.

Tenint en compte que per controlar els dispositius de la xarxa es faran connexions breus i tan sols quan l'usuari ho demani, el BLE encaixa molt bé perquè permet als dispositius tenir una gran autonomia.

### 3.1.2. Wi-Fi

D'entre totes les tecnologies de comunicació sense fil, una de les més usades avui en dia és el Wi-Fi. L'origen del Wi-Fi remunta a finals dels anys 90 quan unes quantes empreses es van associar per crear la Wireless Ethernet Compatibility (WECA), grup que més endavant es va re nomenar com a Wi-Fi Alliance. Aquesta associació tenia com objectiu promoure l'ús del Wi-Fi així com crear uns estàndards per augmentar la compatibilitat entre dispositius.

Amb l'increment de dispositius mòbils com smartphones, tablets i portàtils, l'ús del Wi-Fi per connectar-se a internet ha augmentat exponencialment fins al punt que s'estima que el 60% del tràfic d'internet es fa amb aquesta tecnologia.

Els dispositius Wi-Fi originalment feien ús de la banda de 2.4 GHz tot i que ara s'ha afegit la possibilitat de transmetre a 5 GHz. La diferència entre les dues freqüències és que la baixa pot arribar més lluny i l'alta pot transmetre més informació. Pel que fa a la seguretat, es pot protegir una xarxa Wi-Fi amb una contrasenya anomenada WPA (Wi-Fi Protected Acces).

El Wi-Fi presenta avantatges com per exemple que és econòmic crear-ne xarxes que en facin ús o un gran abast. A més, com a avantatge respecte a el Bluetooth, múltiples dispositius es poden connectar simultàniament a un punt d'accés i també és capaç d'assolir velocitat de comunicacions molt més altes.

Per contra, la seva principal desavantatge si es vol fer servir per internet of things, és el seu elevat consum. Per aquest motiu no és adient per dispositius que funcionant amb bateries i han d'aguantar una temporada llarga.

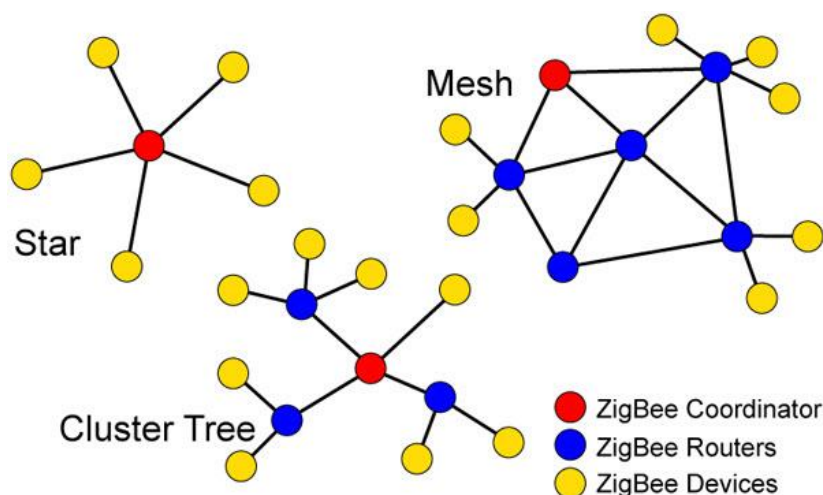
### 3.1.3. ZigBee

És una tecnologia de comunicació per radiofreqüència pensada per intercanviar informació entre dispositius sense un consum gaire elevat.

En les xarxes ZigBee es poden diferenciar 3 tipus de nodes:

- Coordinador: S'encarrega de gestionar la xarxa, iniciant-la, gestionant com es connecten els dispositius entre ells i transferir els paquets de dades. Tota xarxa ZigBee ha de tenir-ne un com a mínim.
- Router: Connecta diferents nodes entre ells, s'utilitza per ampliar l'abast de la xarxa.
- Dispositiu final: Són dispositius que generen o consumeixen les dades que circulen per la xarxa. Només es poden comunicar amb un coordinador o un router i no entre ells.

En la següent imatge es poden veure diferents estructures de xarxes que es poden crear:



*Figura 3.1 Diferents estructures de xarxes ZigBee*

Com a desavantatge cal destacar que no es suporta directament ni per Android ni IOS, és a dir, l'usuari no es pot connectar directament a un dispositiu ZigBee sinó que ha de comunicar-se amb el que fa de node central de la xarxa mitjançant una altra tecnologia com per exemple Wi-Fi.

### 3.1.3. Z-Wave

És un protocol de comunicacions per radiofreqüència que està dissenyat per crear xarxes de dispositius amb una velocitat de transmissió reduïda i a baixa potència. Actualment és usat sobretot en la domòtica.

La diferència principal amb ZigBee és que Z-Wave fa ús d'una freqüència més baixa (908.42 MHz) cosa que permet que pugui arribar a major distància a costa de sacrificar velocitat de transmissió.

Z-Wave presenta com a avantatge destacable que ha estat dissenyat perquè tots els dispositius que l'utilitzen, independentment del fabricant o de l'any de construcció, es poden comunicar entre ells. Per exemple en el cas de Bluetooth, els dispositius amb una versió inferior a la 4.0 no poden comunicar-se amb un que faci ús de LE.

Pel que fa als desavantatges, la major d'elles és la mateixa que per a ZigBee: Cal un dispositiu addicional per comunicar-se amb smartphones o ordinadors degut a que no són compatibles amb aquesta tecnologia.

### 3.1.4. NFC

La Near Field Communication és una tecnologia de comunicació sense fils que prové de RFID amb la diferència que està pensada per a ser utilitzada en distàncies molt curtes (de l'ordre dels cm).

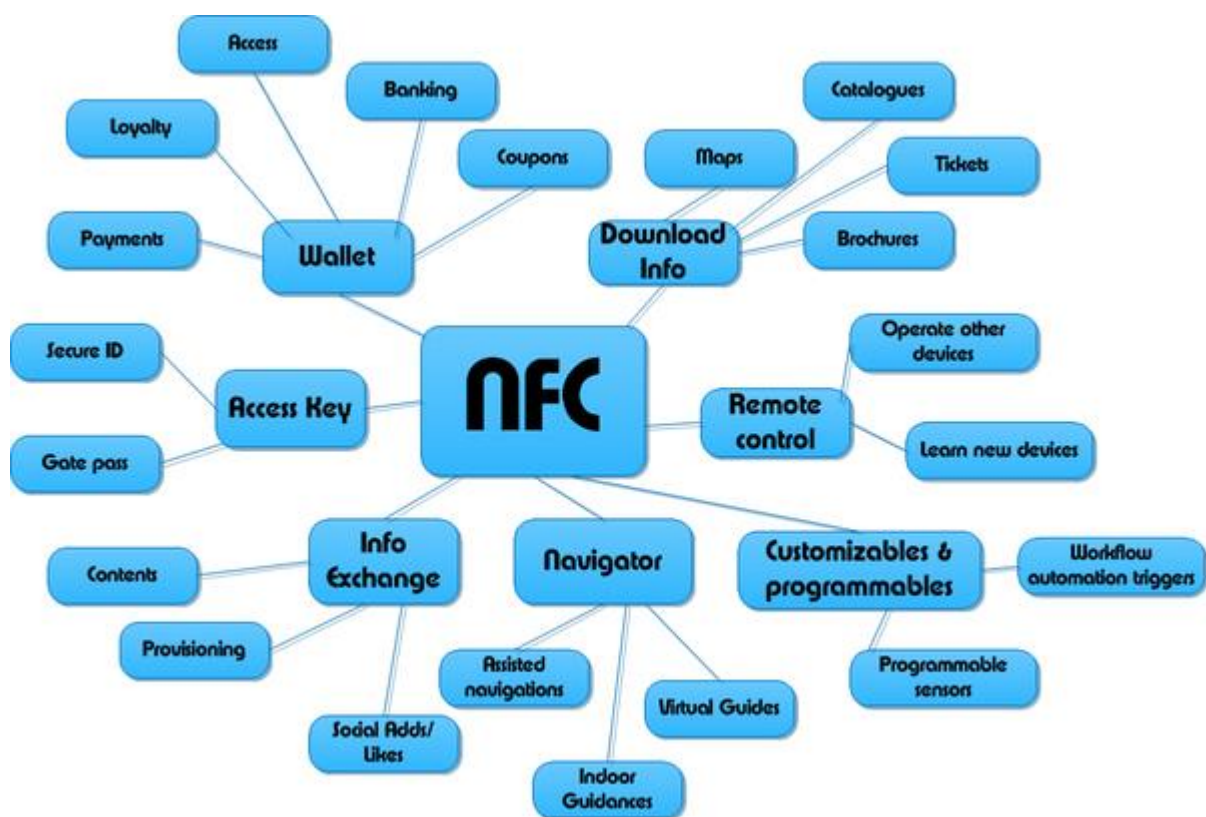
En els últims anys ha crescut el nombre de telèfons mòbils que incorpora aquesta tecnologia degut a que diverses companyies bancàries ofereixen la possibilitat de pagar des del mòbil.

A més també se sol utilitzar per a identificació o bitllets de transport.

Es poden distingir dos modes de funcionament:

- Passiu: No requereix d'alimentació. S'utilitza el camp magnètic creat per a l'altre dispositiu. Un exemple d'aquest mode de funcionament són els tags. Permeten llegir i escriure informació persistent utilitzant un dispositiu actiu.
- Actiu: El dispositiu genera un camp magnètic per tant necessita alimentació. Aquest mode es pot utilitzar per llegir o escriure a altres dispositius així com per actuar emulant un dispositiu passiu.

En aquest projecte, es podria utilitzar la tecnologia NFC amb combinació amb una altra per identificar al dispositiu que es vol afegir a la xarxa.



*Figura 3.2 Diferents usos que se li dóna al NFC actualment*

### 3.1.5. Elecció Final

Fent una comparativa de les tecnologies que hi ha actualment per a dispositius IOT, s'ha decidit que la millor opció per al projecte és fer servir connexions BLE.

El principal motiu per a aquesta elecció a més de pel seu reduït consum és que pràcticament tots els smartphones disposen de BLE cosa que permet controlar els dispositius directament des d'un d'aquests aparells.

Per tal de realitzar una connexió bluetooth, cal conèixer la direcció mac del dispositiu al qual es vol connectar. Això provoca que si es desconeix s'hagi d'escanejar en cerca de dispositius propers i seleccionar el desitjat. Per facilitar la identificació i evitar fer una selecció d'una llista, s'utilitzarà un tag NFC en el que s'haurà gravat la direcció mac del dispositiu en qüestió. Aquests tags són molt econòmics i permeten transmetre informació de forma passiva. Tot i que no tots els telèfons disposen de NFC,



últimament hi ha una tendència a incorporar-lo per poder fer pagaments amb telèfons mòbils.

En resum, en aquest projecte s'utilitzarà Bluetooth LE i NFC per tal de formar la xarxa de dispositius a controlar.

### **3.2. Plataforma d'automatització**

Una plataforma d'automatització per a smarthomes és un conjunt d'eines que permet controlar múltiples dispositius com poden ser llums, calefacció, portes o altres elements de la llar des d'un sol lloc. Aquestes plataformes solen permetre la creació de regles d'automatització, per exemple que el jardí es regui sol si el nivell d'humitat del sòl és massa baix.

En aquest treball s'ha decidit que a més d'oferir la possibilitat de controlar els dispositius des del telèfon mòbil, també es buscarà donar l'opció d'integrar la xarxa creada en una d'aquestes plataformes. En el moment de buscar quina plataforma es podria utilitzar per integrar-se amb el projecte s'ha tingut en compte els següents factors:

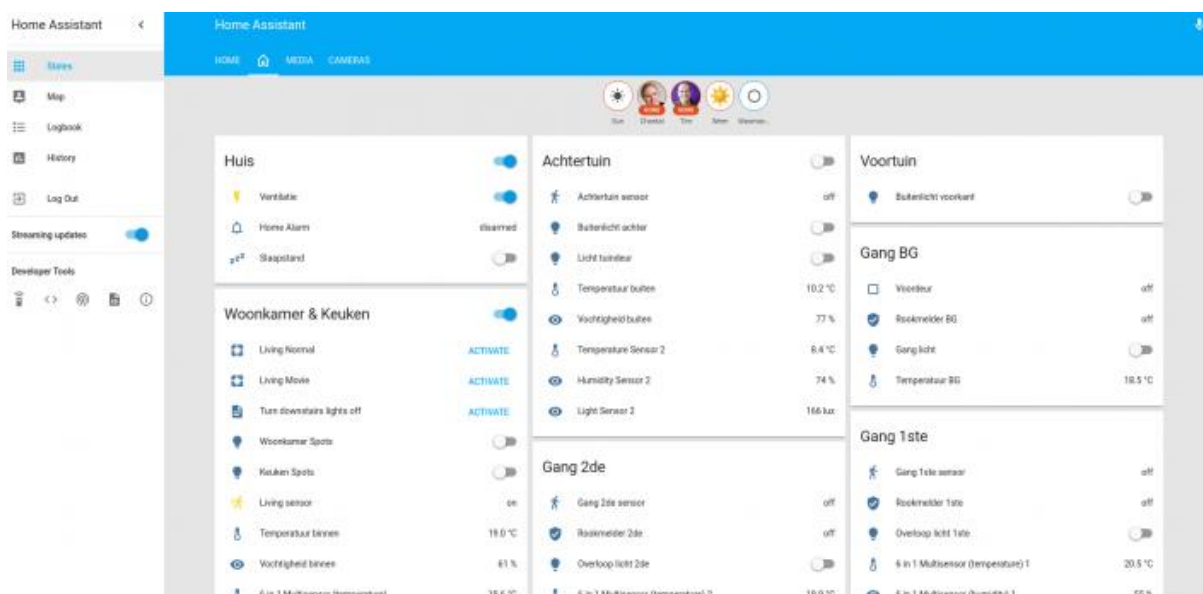
- **Compatibilitat:** La plataforma ha de ser capaç d'acceptar diferents protocols de comunicació a més de permetre afegir dispositius personalitzats i no solament comercials.
- **Nombre d'usuaris:** Per tal d'arribar al major públic possible, tan sols es tindrà en consideració les plataformes amb una comunitat d'usuaris més grans.

D'entre les diferents opcions que hi ha disponibles s'ha analitzat les més populars: Home Assistant i OpenHab.

### 3.2.1. Home Assistant

L'any 2014, un enginyer de software anomenat Paulus Schoutsen va comprar unes bombetes intel·ligents que es controlaven per wifi. Per automatitzar el control d'aquestes bombetes, va crear un petit programa amb el que les podia controlar des d'un navegador web. Un cop fet això va augmentar la intel·ligència del sistema seguint la posició del sol i amb un detector de presència. Si el sensor detectava que era a casa i la posició del sol era baixa, les llums s'encenien soles. Un cop el projecte va madurar, el seu creador va decidir compartir-lo de forma gratuïta penjant el seu codi sota una llicència open source. Avui en dia, aquest petit projecte al qual va posar com a nom Home Assistant, porta més de 100 actualitzacions i té una important comunitat d'usuaris.

El control dels components es fa accedint amb un navegador web al dispositiu on Home Assistant s'està executant i que actua com a servidor web. En la següent imatge es pot veure un exemple de panell de control amb múltiples dispositius configurats.



*Figura 3.3 Panell de control de HA*

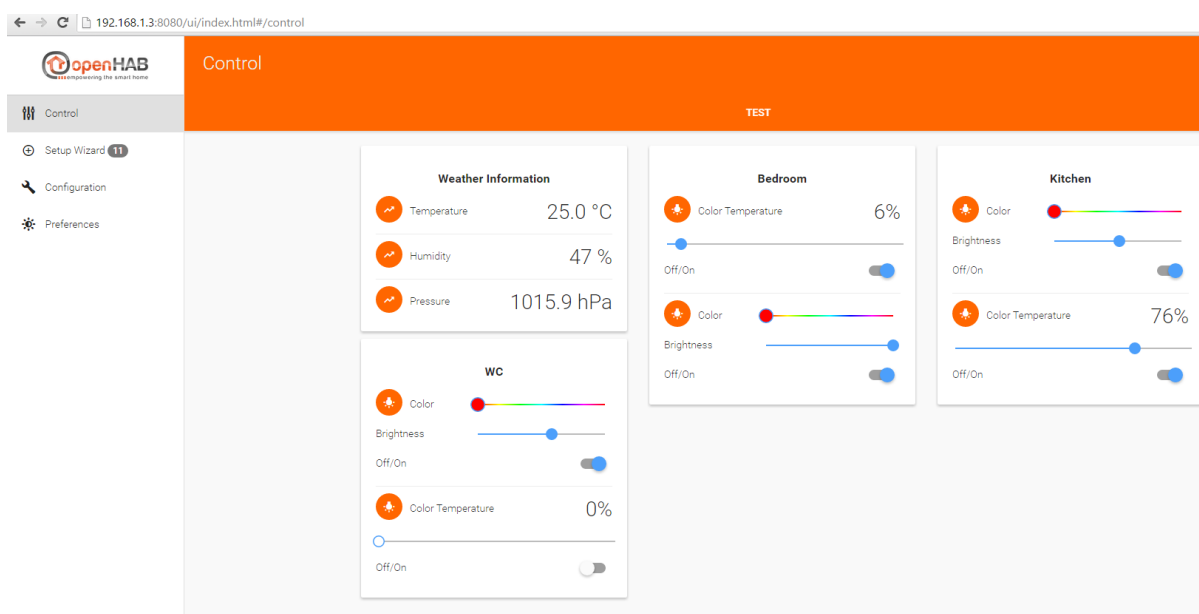
Aquesta plataforma implementa millores constants a bon ritme, per exemple en una actualització recent es dona la possibilitat de crear regles d'automatització directament des del panell de control, sense necessitat de modificar el fitxer de configuració.

### 3.2.2. OpenHab

OpenHab està dissenyat per funcionar de forma modular, és a dir, existeix un nucli central i d'altres funcionalitats que s'activen d'acord amb els components que es vulguin fer servir. Tot i aquesta arquitectura, consumeix més recursos que les altres alternatives.

L'organització dels agents que intervenen en la plataforma és la següent:

- Things: Són les entitats que es poden afegir a la plataforma. Aquests s'utilitzen per obtenir informació o controlar dispositius. Poden ser objectes, com per exemple un relé o serveis informàtics com podria ser un api web.
- Items: Sensors o actuadors que són controlats per les things com per exemple una bombeta.
- Channels: El mitjà a través del qual l'usuari es comunica amb les things per tal de controlar-les.
- Links: Les unions entre els items i les things, per exemple un cable en cas que siguin dispositius físics.



*Figura 3.4 Panell de control de OH*

Després de dos anys de versions beta, a principis de 2017 van llançar la versió 2.0. Amb aquesta actualització, a més de correccions i optimitzacions, s'ofereixen noves funcionalitats.

Per facilitar-ne el control des d'un smartphone, disposa d'aplicacions oficials tant per IOS com Android.

### **3.2.3. Comparativa**

Després d'analitzar les dues opcions, s'ha decidit utilitzar Home Assistant pels següents motius:

- Velocitat d'actualitzacions: Home Assistant (HA) destaca per adoptar ràpidament noves tecnologies així com nous productes. Degut a que té actualitzacions freqüents, els desenvolupadors poden anar integrant nous components a mesura que aquests es llancen al mercat.
- Comunitat d'usuaris: Hi ha un gran nombre de persones que comparteixen les seves configuracions de HA cosa que redueix la complexitat de la integració de nous components.
- Control de les automatitzacions: HA permet apagar o encendre les automatitzacions que es fan amb els components directament des del panell de control, sense necessitat de tocar la configuració.
- Historial: Es pot consultar l'historial d'ús de tots els components que s'utilitzen. Això pot ser útil per exemple per veure quant de temps ha estat encesa una llum.
- Dificultat d'ús: HA està pensat perquè sigui senzill d'utilitzar. Es pot afegir un component tan sols modificant un parell de línies al fitxer de configuració.
- Documentació: En la web del projecte, es pot trobar una documentació completa de tot el que es pot fer amb aquesta eina. A més d'una explicació de cada cosa, hi ha diferents exemples en cada cas.
- Interfície: La UI de HA és agradable, intuïtiva i s'adapta a qualsevol resolució, ja sigui d'ordinador, tablet o telèfon.
- Consum de CPU: HA requereix menys recursos i es pot instal·lar sense cap problema en dispositius de baix rendiment com una raspberry pi.

## 4. Disseny

### 4.1. Agents del sistema

Un cop seleccionades les tecnologies i plataformes amb les quals s'implementarà el projecte, s'ha procedit a dissenyar-ne l'arquitectura. Partint de les especificacions, s'han definit dos agents als quals s'ha anomenat Network Manager i Home Nexus.

#### 4.1.1. Network Manager

La primera acció que haurà de fer l'usuari per crear la xarxa, serà identificar quins dispositius la formen. Per fer això, utilitzarà un smartphone en el qual haurà instal·lat una aplicació que s'ha batejat com a Network Manager.

Apropant el smartphone a l dispositiu que es vulgui controlar, Network Manager llegirà un tag NFC on hi ha enregistrada la direcció MAC d'aquest i en cas que sigui el primer cop que es llegeix, es procedirà a configurar el dispositiu.

La configuració consisteix en un seguit de comandes amb les quals s'obté informació sobre quants sensors / actuadors disposa i de quin tipus són. Un cop creada la xarxa, apareixeran a l'aplicació els diferents actuadors i sensors que s'hagin configurat.

Tal com s'ha esmentat a les especificacions del projecte, es donarà l'opció a usuaris més avançats de transferir la xarxa a la plataforma d'automatització anomenada Home Assistant. Aquesta comunicació amb Home Assistant serà responsabilitat d'un altre agent que s'ha anomenat Home Nexus. L'única comunicació que es realitzarà entre Network Manager i Home Nexus serà quan l'usuari vulgui afegir o eliminar dispositius de Home Assistant i consistirà en enviar una descripció de la xarxa.

Després de definir les funcions de Network Manager, s'han distribuït en els següents mòduls per tal d'estructurar-ne el desenvolupament:

- User Interface: És el conjunt de pantalles que formen l'aplicació i permeten interactuar amb aquesta fent ús de botons.
- Device Manager: Serà l'encarregat de realitzar les connexions Bluetooth amb els dispositius, generar els missatges a enviar en funció dels inputs de l'usuari i processar la resposta d'aquests.

- Data Manager: Per tal d'evitar que l'usuari hagi de configurar un dispositiu cada cop que el faci servir, s'emmagatzemarà les dades necessàries per realitzar les connexions Bluetooth.
- Home Nexus Bridge: S'utilitzarà per compartir la informació de la xarxa amb Home Nexus en el moment en què l'usuari vulgui actualitzar els dispositius que té actualment a Home Assistant.

Per facilitar la comprensió de l'estructura de Network Manager, s'ha fet el següent esquema on es poden veure els diferents mòduls que el formen, i com es comunica amb els altres:

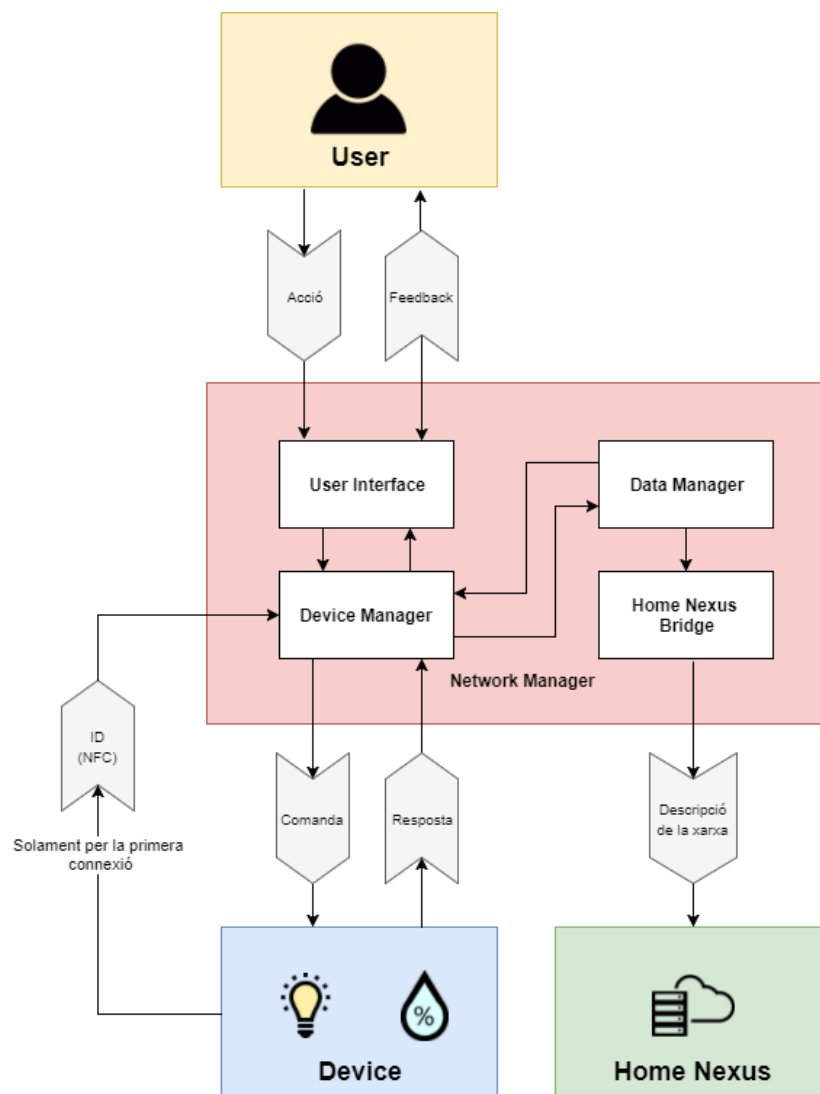


Figura 4.1 Disseny de Network Manager

### 4.1.2. Home Nexus

Aquest mòdul serà utilitzat per fer de pont entre els dispositius a controlar i Home Assistant. En concret permetrà a l'usuari realitzar dos classes d'accions:

- **Traspàs de la xarxa:** Quan l'usuari selecciona enviar la xarxa des de Network Manager, s'utilitza aquest mòdul com a pont entre Home Assistant.
- **Control de la xarxa:** Quan l'usuari vol actuar sobre algun dels dispositius, ho comunicarà a aquest mòdul i serà l'encarregat d'interactuar amb el dispositiu objectiu.

El seu desenvolupament s'ha dividit en els següents mòduls:

- **Device Manager:** És el nucli de Home Nexus. Aquest element és l'encarregat de connectar-se per bluetooth a un dispositiu quan és necessari, enviar les dades i tractar la resposta.
- **Home Assistant Bridge:** Aquest submòdul és l'encarregat de comunicar-se amb Home Assistant. Al iniciar-lo es connecta al servidor que ve per defecte amb HA i rep els missatges que es generen cada vegada que l'usuari vol controlar algun actuator. Pel que fa als sensors, cada cop que device manager li passi una dada d'una lectura l'enviarà a HA.
- **Network Manager Bridge:** És un servidor HTTP que s'encarrega de comunicar-se amb la app per a smartphones. Aquesta comunicació tan sols es produeix quan la app ho sol·licita i s'utilitza per afegir els dispositius que té la app amb els de Home Assistant.
- **Data Manager:** S'encarrega de emmagatzemar els dispositius de la xarxa en una base de dades per recuperar-los si es reinicia la aplicació. Per afegir un dispositiu a HA, caldrà modificar el seu fitxer de configuració.

En el següent esquema es poden veure els diferents mòduls que formen Home Nexus i com es comunica amb els altres elements del sistema.

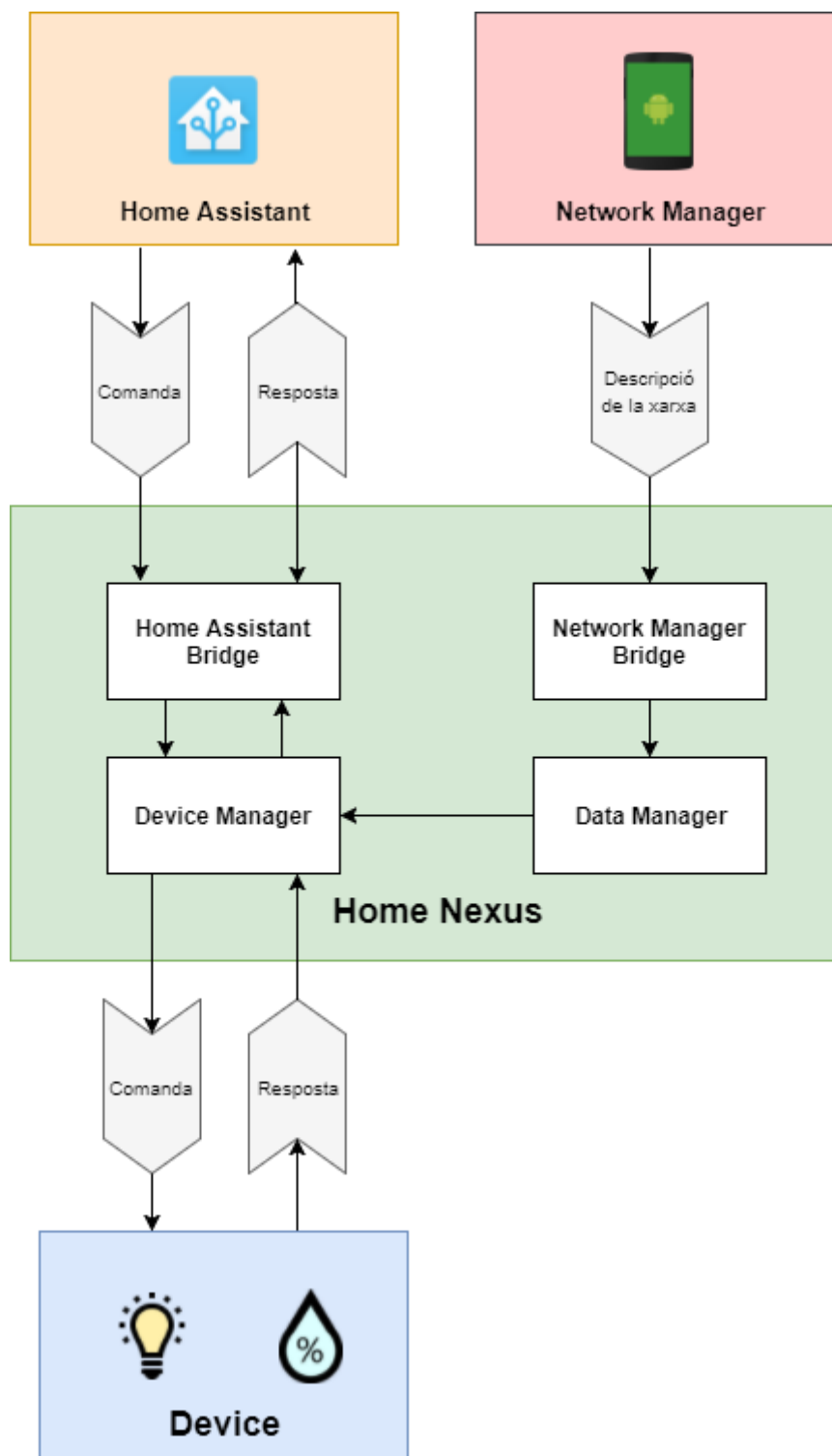


Figura 4.2 Disseny de Home Nexus



## 4.2. Dispositius prototip

En el transcurs del treball, caldrà preparar un prototip que simuli el comportament d'aquests dispositius per tal de validar el sistema. En concret, s'utilitzaran microcontroladors juntament amb sensors, actuadors i mòduls de Bluetooth per representar dispositius comercials que afegiria l'usuari a la seva xarxa. Per identificar-los amb el smartphone s'utilitzarà una etiqueta NFC per a cadascun.

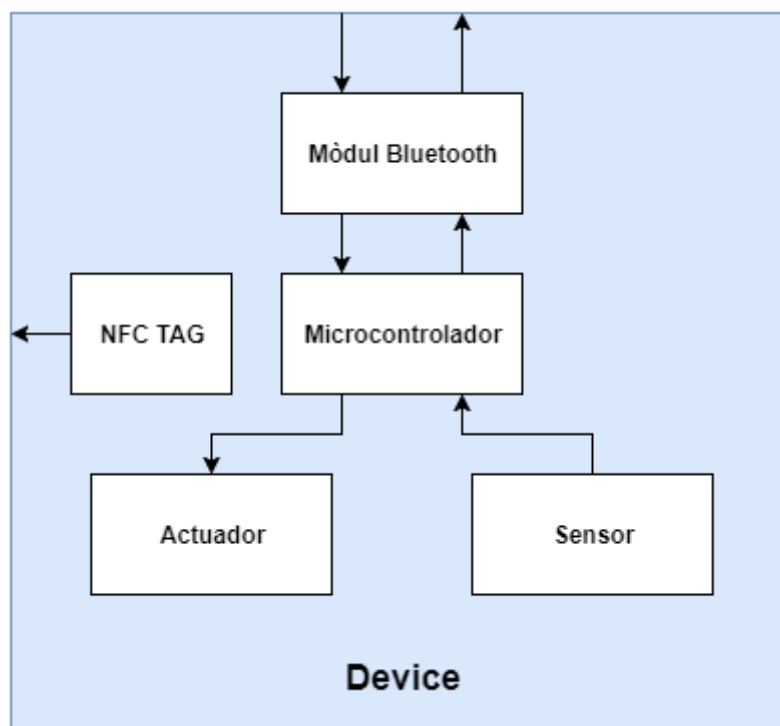
Es pretén que aquest dispositiu pugui actuar de forma genèrica independentment del tipus d'actuador o sensor que es vulgui connectar, en concret s'oferirà la possibilitat d'encendre o apagar els actuadors i controlar-ne els sensors.

### 4.2.1. Components del dispositiu

El dispositiu estarà format per els següents elements:

- Un mòdul Bluetooth que serà l'encarregat de rebre les comandes i transmetre les respostes.
- Un microcontrolador que serà l'encarregat de processar les comandes, actuar en conseqüència i transmetre la resposta al mòdul Bluetooth.
- Sensors i actuadors que són els dispositius amb els que l'usuari vol interactuar.

En el següent esquema, es pot veure com estan estructurats aquests components:



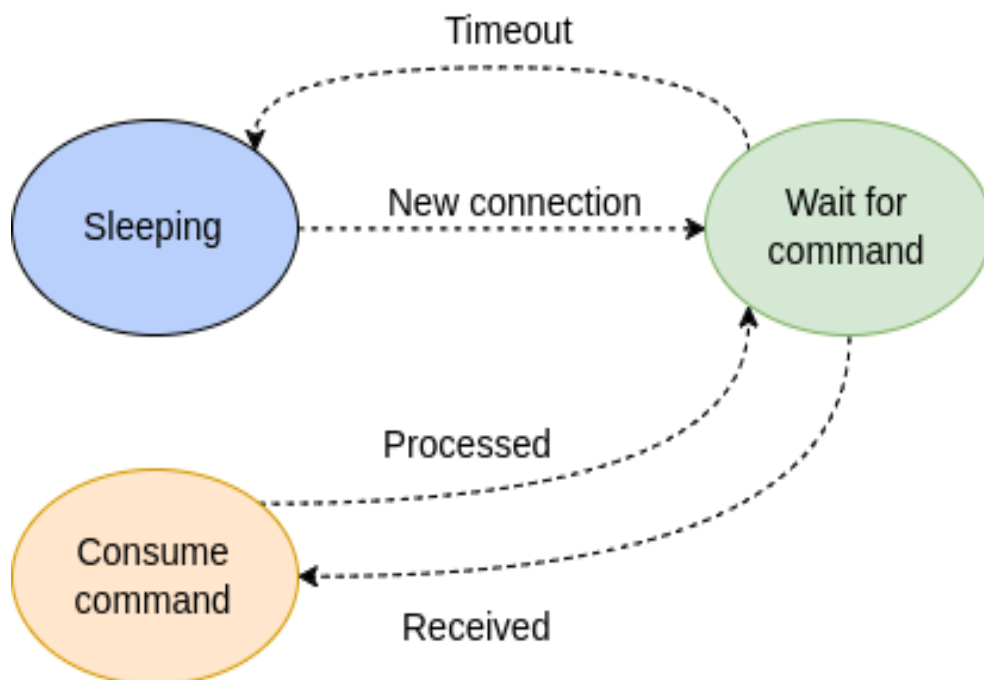
*Figura 4.3 Disseny del dispositiu*

### 4.2.2. Estats del dispositiu

El prototip disposarà de 3 estats de funcionament:

- **Sleep:** El dispositiu està en baix consum a l'espera d'una connexió. En aquest estat el microcontrolador està dormint i tan sols pot ser despertat per una interrupció provocada per una sol·licitud de connexió. Pràcticament la totalitat del temps de funcionament es trobarà en aquest estat.
- **Waiting command:** Un cop es rep una connexió al mòdul Bluetooth, el microcontrolador rep una interrupció que fa que es desperti i es posi a l'espera d'instruccions. En cas que passin 10 segons sense una comanda vàlida es torna a dormir.
- **Consuming command:** El microcontrolador processa la comanda rebuda i envia una resposta. Si la comanda és vàlida es reinicia el contador de wait for command. Un cop enviada la resposta es torna a l'estat waiting command.

En el següent diagrama es mostren els tres estats així com les transicions:



*Figura 4.4 Màquina d'estats del prototip*

## 5. Implementació

### 5.1. Home Nexus

Com s'ha comentat en l'apartat del disseny, Home Nexus és un programa transparent a l'usuari que es fa servir de pont entre els dispositius a controlar i Home Assistant. Es va decidir utilitzar el llenguatge Python per implementar-lo pels següents motius:

- Home Assistant està implementat amb Python.
- És un llenguatge fàcil d'aprendre i el codi resultant es molt llegible, un avantatge si algú amb pocs coneixements d'informàtica decideix fer alguna modificació.
- És possible implementar els quatre submòduls definits en el disseny degut a que existeixen llibreries tant per a bluetooth, MQTT i HTTP.

#### 5.1.1. Home Assistant Bridge

És la part del programa encarregada de comunicar-se amb Home Assistant. Aquesta comunicació es pot iniciar per dos motius: Quan l'usuari demana controlar un actuator des del panell de Home Assistant o de forma automàtica cada cert temps per mirar el valor dels sensors.

Per intercanviar informació entre HA i aquest submòdul, s'utilitza el protocol de comunicació MQTT (Message Queue Telemetry Transport). Aquest protocol funciona sobre TCP i es basa en la subscripció / publicació de missatges. Està pensat per connexions amb poc ample de banda pel que s'ha dissenyat perquè els missatges siguin breus. Per aquest motiu es bastant utilitzat en dispositius IOT. A continuació es mostra els mètodes que es faran servir:

- **Connect:** Inicia una connexió al servidor.
- **Subscribe:** Es subscriu a un topic. Un cop fet rebrà els missatges que circulin per aquest.
- **UnSubscribe:** Es dona de baixa del topic per deixar de rebre'n missatges.
- **Publish:** Escriu un missatge a un topic. Tots els clients subscrits a aquest el rebran.
- **Disconnect:** Finalitza la connexió al servidor.

Per entendre millor el seu funcionament, es prendrà com a exemple una xarxa on hi ha tres dispositius connectats a un altre que actua com a broker. Primer de tot els clients es connectaran al broker. A continuació, es subscriuran als temes als quals volen rebre o publicar informació (Subscribe temperature2). Finalment quan un client vulgui compartir una dada, farà un publish al tema (Publish temperature2 29) i tots els que s'hi hagin subscrit rebran les dades.

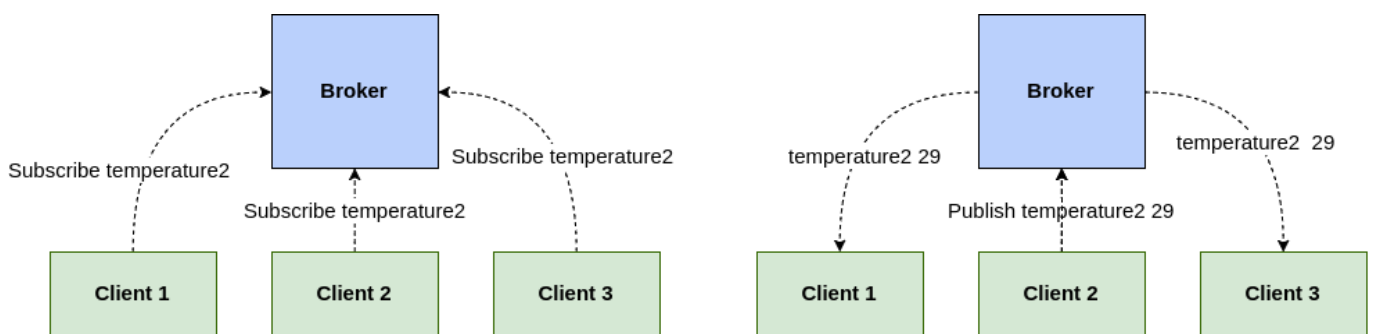


Figura 5.1 Exemple de xarxa MQTT

Home Assistant Bridge també serà l'encarregat d'afegir o eliminar nous dispositius quan l'usuari ho demani amb el seu smartphone. Per modificar els dispositius que utilitza HA caldrà escriure al seu fitxer de configuració. Aquest fitxer emmagatzema les dades en un fitxer en format YAML (un format pensat per facilitar la lectura) anomenat "configuration.yaml". És important saber que per iniciar els dispositius aquest fitxer només es llegeix al arrancar HA, per tant si es modifica per tal d'afegir o eliminar dispositius caldrà reiniciar-lo. A continuació es mostra com es representarà un sensor en aquest format:

```
- platform: mqtt
  name: 001583407A5B_Humidity1
  state_topic: "00:15:83:40:7A:5B/dev1/Humidity1"
```

Cal tenir en compte que tan el nom que se li dona al dispositiu com el tòpic tenen que ser únics. Per evitar conflictes, s'usarà la direcció mac i el nom per defecte del dispositiu per formar-los. Com que queda una mica confús, s'utilitzarà una funcionalitat anomenada customize que permet modificar el nom amb el que

apareixen els components. Serà en aquest camp on s'usarà el nom donat per l'usuari al dispositiu des de l'aplicació. A continuació es mostra com es modifica el nom:

*sensor.001583407A5D\_Humidity1:*

*friendly\_name: Humidity2*

En el cas dels actuadors, s'ha decidit utilitzar el component switch. Aquest apareix com un boto commutador al panell de Home Assistant amb el qual l'usuari pot encendre o apagar el dispositiu. La configuració d'un actuator seria la següent:

*- platform: mqtt*

*name: 001583407A5D\_Led1*

*state\_topic: "00:15:83:40:7A:5D/dev1/Led1"*

*command\_topic: "00:15:83:40:7A:5D/dev1/Led1/set"*

Es diferencia amb els sensors en que hi ha dos topics MQTT, state per informar de l'estat en que es troba l'actuator i command per controlar-lo.

En resum, el mòdul Home Assistant Bridge és l'encarregat de:

- Iniciar Home Assistant.
- Afegir i eliminar dispositius al fixer de configuració yaml.
- Controlar les comunicacions MQTT entre HA i Home Nexus.
- Llegir els sensors cada cert temps.

### 5.1.2. Network Manager Bridge

Quan l'usuari desitgi afegir o eliminar dispositius de HA, caldrà comunicar la aplicació mòbil amb Home Nexus. Per fer això s'utilitzarà un servidor HTTP juntament amb el mètode POST. Aquest servidor s'executarà en un thread apart per estar disponible en tot moment. En el moment que es detecti una petició POST es reuperaran les dades de la xarxa en format JSON. S'han definit 3 modes d'interacció entre Home Nexus i Network Manager:

- **Update:** Si es rep el mode update, s'actualitzaran diferents dades dels dispositius que ja té registrat Home Nexus tals com el noms o els subdispositius dels quals disposa. En cas que no estigui un dispositiu a la xarxa, se l'afegirà.
- **Replace:** En cas de rebre una petició per substituir la xarxa, s'eliminaran els dispositius que hi ha a Home Nexus i se'ls substituirà amb els que arribin per la petició POST.
- **Delete:** S'eliminaran tots els dispositius de Home Nexus.

### 5.1.3. Device Manager

Alhora de representar un dispositiu en el programa, s'ha decidit diferenciar dues classes: Dispositius i subdispositiu.

- **Dispositius:** Són els aparells als quals l'aplicació es connecta per Bluetooth per tal de controlar els subdispositius dels quals disposa. Un dispositiu pot tenir un o més subdispositius.
- **Subdispositius:** Són els sensors i actuadors que vol controlar l'usuari. L'usuari no es pot comunicar directament a través d'ells si no que utilitza el dispositiu pare de cada subdispositiu com a intermediari. Per distingir si un subdispositiu és un actuator o un sensor, s'utilitza el tipus de subdispositiu, dada que s'ha obtingut en el procés de configuració.

#### 5.1.4. Data Manager

Per evitar la pèrdua de les dades que representen els dispositius, caldrà emmagatzemar-les. En concret cada cop que es rebí una comunicació des de l'aplicació mòbil, es passaran a aquest mòdul per tal que les emmagatzemi en una base de dades sqlite.

Es necessitaràn les següents taules per emmagatzemar les dades:

- **Devices:** id, defaultName, customName, mac.
- **Subdevices:** id, defaultName, customName, type, parentDevice.
- **Sensor Logs:** id, value, timestamp, parentSensor.

### 5.2. Network Manager

Degut a que la duració del projecte no es molt extensa, es va decidir implementar la app per a smartphones tan sols en el sistema operatiu Android. Per desenvolupar aquesta app, s'ha utilitzat el llenguatge Java i l'entorn de desenvolupament Android Studio. A continuació s'explicarà com s'utilitza l'aplicació per afegir dispositius, controlar-los i enviar la informació a Home Assistant.

#### 5.2.1. Addició de dispositius

Quan l'usuari obre l'aplicació, troba tres pestanyes en les quals apareixen els dispositius, actuadors i sensors que hi ha a la xarxa. En el primer ús, en aquestes pestanyes apareixerà un missatge que indica que cal apropar el telèfon al dispositiu que es vol afegir a la xarxa per tal que el sensor NFC el detecti. En el moment en què es llegeixi el Tag, l'aplicació es connectarà automàticament a aquest dispositiu per bluetooth i procedirà a configurar-lo. En aquest procés de configuració, es pregunta quants subdispositius té aquell dispositiu (Sensors i actuadors) i quin nom i tipus tenen. A continuació es mostra un exemple d'aquest procés:

Generador	Comanda	Significat
Network Manager	SDEV	Pregunta pel nombre de subdispositius
Dispositiu	SDEV 2	Respon que té 2 dispositius
Network Manager	SDEV1	Demana informació del dispositiu 1
Dispositiu	SDEV1 Humidity1 1	Respon que es diu Humidity1 i que es de tipus 1
Network Manager	SDEV2	Demana informació del dispositiu 2
Dispositiu	SDEV2 Led1 1	Respon que es diu Led1 i que es de tipus 1

Un cop configurat el dispositiu, apareix juntament amb tots els subdispositius que tingui a cada pestanya tal com mostren les següents imatges:

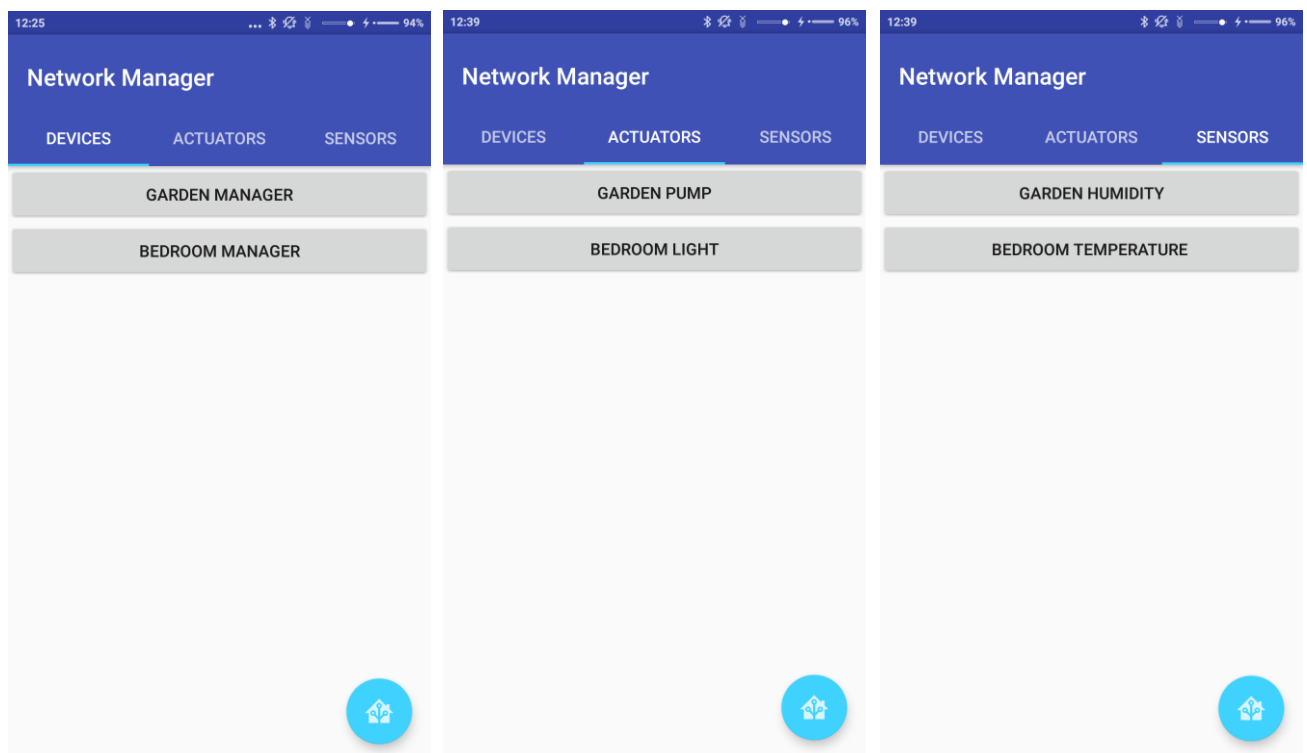


Figura 5.2 Selecció de dispositius amb Network Manager



### 5.2.2. Control de dispositius

Quan l'usuari vol controlar un dispositiu cal que el seleccioni en la pestanya corresponent. Segons el tipus del subdispositiu se sap si és un actuator o un sensor. Amb aquesta informació es poden diferenciar dos tipus d'interacció: Encendre o apagar un actuator i llegir les dades dels sensors. A més es permet a l'usuari modificar el nom amb què veu el dispositiu (cal distingir aquest nom de l'utilitzat per controlar-lo, el qual no es pot modificar).

Si l'usuari desitja eliminar el dispositiu, ho pot fer eliminant el dispositiu pare o cadascun dels subdispositius que hi ha.

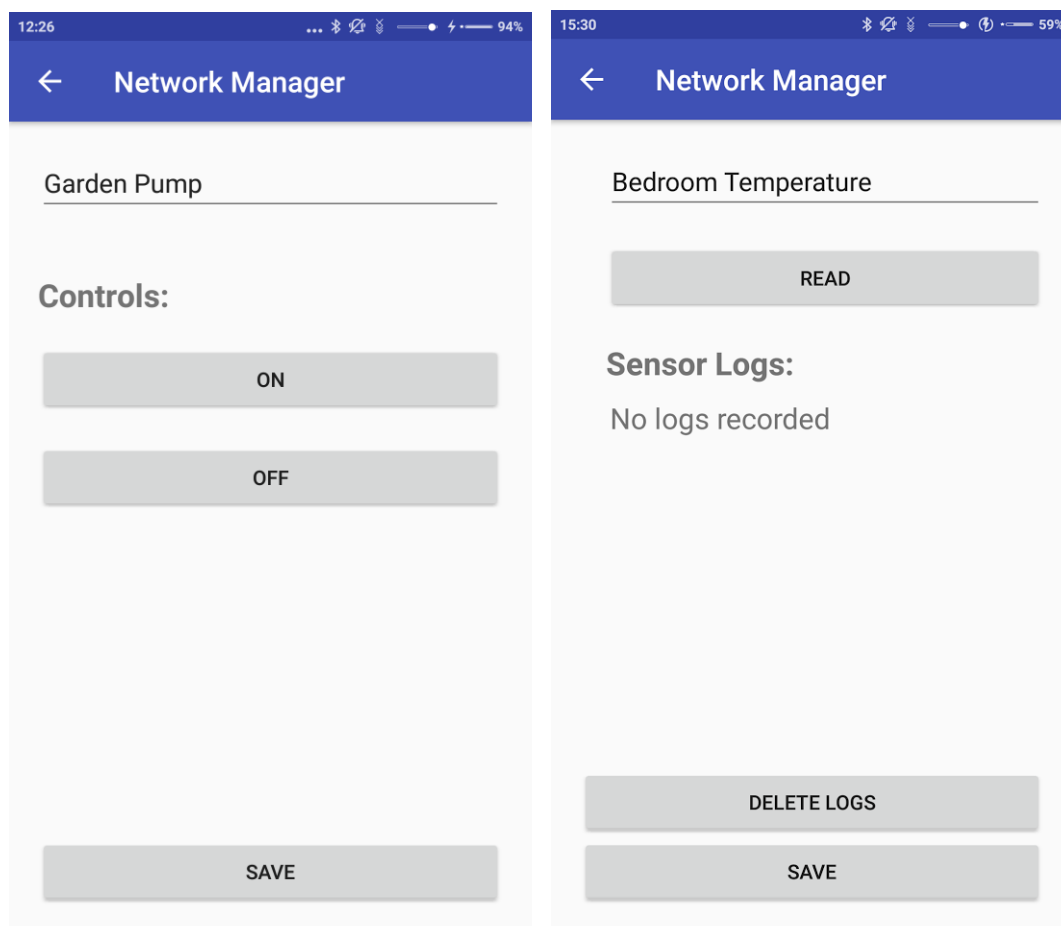


Figura 5.3 Control dels dispositius

### 5.2.3. Interacció amb Home Assistant

Fent ús del botó flotant que apareix en les pestanyes de selecció dels dispositius, es pot accedir a la pantalla d'interacció amb Home Assistant. En aquesta pantalla, l'usuari haurà d'introduir la direcció Http del seu servidor HA per tal que l'aplicació s'hi pugui comunicar. Un cop introduïda es pot guardar per a un ús posterior així com modificar-la en qualsevol moment.

Per afegir els dispositius a la xarxa HA, es donen tres opcions:

- **Update:** Actualitzar la informació dels dispositius que hi ha a HA amb les dades de l'aplicació.
- **Replace:** Actualitzar la informació dels dispositius que hi ha a HA amb les dades de l'aplicació a més d'eliminar els que actualment no es trobin a l'aplicació.
- **Delete:** Eliminar tots els dispositius de Home Assistant afegits amb l'aplicació.

En prémer un dels 3 botons, es genera una petició HTTP POST a la direcció HTTP afegida prèviament. Si aquesta *url* correspon a Home Assistant, el submòdul Network Manager Bridge de Home Nexus rebrà la petició i realitzarà les accions corresponents.

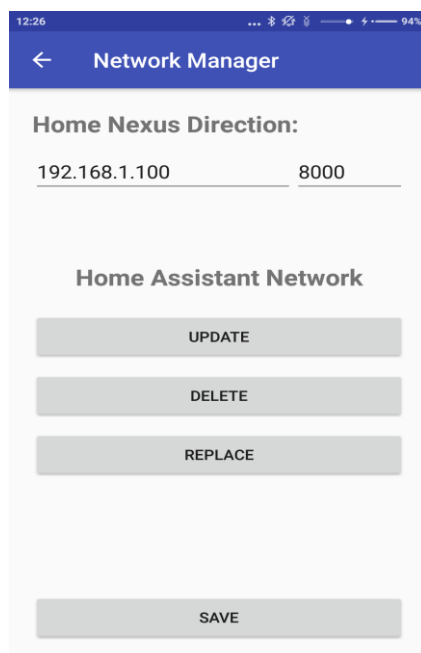


Figura 5.4 Connexió amb Home Assistant

Si la petició ha tingut èxit, es mostrarà un missatge a la pantalla i al accedir a Home Assistant es podrà veure que en el panell han aparegut els diferents components que s'han enviat.

### **5.3. Interacció entre dispositius**

Un cop afegits els dispositius a Home Assistant gràcies a Home Nexus i l'aplicació, l'usuari es pot aprofitar de l'automatització que ofereix HA per crear interaccions entre dispositius.

Abans de la versió 0.45 de Home Assistant calia modificar l'arxiu de configuració per tal d'afegir les automatitzacions. Ara s'ofereix la possibilitat d'establir les relacions entre components directament des del panell de control. Per exemple, en el cas que es vulgui iniciar un actuator com a resposta a la lectura d'un sensor caldrà seguir els següents passos:

1. Entrar a l'editor d'automatitzacions en el panell de Home Assistant i crear-ne una nova.
2. Crear un o diversos Triggers i seleccionar el tipus. Hi ha diferents tipus de trigger com timers, lectures d'un topic MQTT o canvis de valors en sensors. Un cop seleccionat el tipus de trigger, cal dir quina condició el farà saltar, per exemple quan una dada sigui superior a cert valor.
3. Crear una o varies accions que es duran a terme quan salti el trigger com podria ser canviar l'estat d'un sensor o enviar una notificació.
4. Guardar l'automatització i activar-la des del panell principal.

## 6. Integració i validació del sistema

En aquest apartat s'explicarà com s'ha implementat el prototip que s'ha fet servir per simular els aparells comercials que l'usuari pot integrar a la seva xarxa.

### 6.1. Dispositius Prototip

Per comprovar que el sistema tingui un funcionament correcte, caldrà desenvolupar un prototip que simuli els dispositius a controlar. Alhora de construir-lo es tindrà en compte els següents factors:

- Haurà de tenir com a mínim un sensor.
- Haurà de tenir com a mínim un actuator.
- Es buscarà un consum reduït per tal que duri el major temps possible.

#### 6.1.1. Hardware

El Hardware que compon el dispositiu està compost per un Microcontrolador, un mòdul Bluetooth, actuadors i sensors, una bateria i un Tag NFC. A continuació s'explicarà la funció de cadascun.

##### Microcontrolador

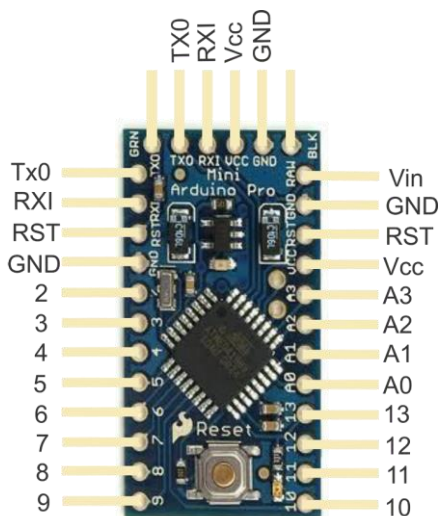


Figura 6.1 Arduino pro mini

Quan el telèfon enviï una comanda Bluetooth, caldrà que el dispositiu la interpreti i actuï en conseqüència. Aquesta tasca la realitzarà un microcontrolador.

Un microcontrolador és un circuit integrat format com a mínim per una CPU, una memòria de programa i una memòria de dades. Els microcontroladors estan presents en molts aparells d'ús diari com poden ser els telèfons, cotxes o electrodomèstics. En ser utilitzats en dispositius tan diferents, n'existeix una àmplia gamma amb prestacions que s'adapten a les necessitats de cada aparell.

En aquest projecte, tan sols es farà servir per interpretar les comandes Bluetooth, llegir sensors i activar actuadors. Com

que aquestes tasques no són gaire complexes, es preveu que la majoria de microcontroladors de la gamma més baixa seran suficients.

## Arduino

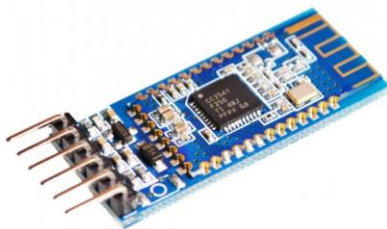
Arduino és una plataforma electrònica open source formada per hardware, software i un entorn de programació que possibilita programar muntar prototips de circuits electrònics de forma fàcil i econòmica.

D'entre el gran nombre de models que ofereix la plataforma, s'ha escollit l'anomenat Pro Mini, que tot i ser un dels més bàsics serà suficient per al que es vol fer servir.

L'Arduino Pro Mini es distingeix per tenir una mida i consum reduïts. Disposa d'un microcontrolador basat en l'Atmel ATmega328 amb una freqüència de rellotge de 8 o 16 MHz i treballa amb un voltatge de 3.3 o 5V (segons la variant). Pel que fa a l'alimentació, com que disposa d'un regulador intern, se li pot subministrar fins a 12V. Segons el comportament descrit en l'apartat de disseny, el microcontrolador es trobarà en la major part de temps en un estat d'hibernació profunda. En aquest mode de repòs s'ha mesurat que el consum és tant sols de 0,2 mA cosa que és molt reduït i permetrà que el prototip aguantí en funcionament setmanes.

## Mòdul Bluetooth

Donat que el microcontrolador escollit no disposa de connexió Bluetooth, caldrà un mòdul extern amb aquesta tecnologia. En el mercat hi ha una àmplia gamma de mòduls compatibles amb Arduino que utilitzen la comunicació sèrie per interactuar amb el microcontrolador. D'entre ells s'ha escollit el model AT-09 que pot ser utilitzat tant com esclau com de mestre però en aquest projecte s'utilitzarà en mode esclau degut a que serà un altre dispositiu (Home Nexus o Network Manager) el que iniciarà la connexió. Quan està connectat i transmetent informació consumeix uns 8 mA, però per reduir aquest consum disposa d'un mode d'auto-sleep en què entra automàticament quan es reinicia o quan se li sol·licita amb una comanda. En aquest mode tan sols consumeix 0.6 mA fins que algun dispositiu s'hi connecta i el desperta.



*Figura 7.2 AT-09*

## Actuadors i sensors

Són els dispositius que es busca controlar. Com s'ha definit en les especificacions, en el cas dels actuadors s'oferirà la possibilitat d'encendre i apagar-los. En el cas dels sensors es podrà llegir el seu valor. En el prototip d'exemple s'ha fet servir un llum led amb el qual es pot visualitzar l'estat de l'actuador (Encès o apagat) i un sensor d'humitat.

## Alimentació

El circuit que es vol alimentar no té un consum gaire elevat. Com que tot funciona a 3,3 volts, s'ha decidit utilitzar una pila recarregable 18650. Amb un multímetre, s'ha mesurat un consum de 0,75 mA quan el prototip està a l'espera de ser connectat. Hi ha una àmplia gamma de capacitats per aquest tipus de bateries. La que s'ha escollit té una capacitat de 2000 mAh cosa que amb el consum mesurat permetrà que el prototip funcioni durant uns 110 dies sense necessitat d'una nova càrrega.

## NFC Tag

Un tag NFC és un xip que s'utilitza per emmagatzemar una petita quantitat d'informació. L'avantatge principal dels tags és que són dispositius passius, és a dir, no requereixen alimentació per funcionar. Quan s'apropa un aparell NFC actiu, aquest genera un camp electromagnètic que el tag utilitza per alimentar-se i generar una resposta en forma d'ona electromagnètica amb la informació que tenia emmagatzemada.



*Figura 7.3 Tag NFC*

Existeixen diferents tipus de tags, alguns tan sols es poden llegir i altres permeten modificar la informació que tenen emmagatzemada.

El tag NFC serà utilitzat per emmagatzemar la direcció MAC del mòdul Bluetooth per tal que al apropar el telèfon amb l'aplicació Network manager, es llegeixi i es pugui connectar per Bluetooth al dispositiu automàticament.

### **6.1.2. Software**

Per programar els microcontroladors Arduino es fa ús d'un llenguatge propi molt semblant a C++ que deriva de processing. L'entorn de programació que ofereix la plataforma permet carregar el software que es dissenya a les plaques de forma fàcil i ràpida.

Tal com s'ha dit en l'apartat de disseny, el microcontrolador es pot trobar en 3 estats: En repòs, processant una comanda o esperant la següent. A continuació es descriu cada una de les funcions dissenyades per implementar el software del prototip:

#### **Setup**

Tot programa d'Arduino disposa com a mínim de dues funcions: Setup i Loop. La funció setup es crida un sol cop just en el moment en què s'inicia el programa. En aquesta funció s'han definit quins pins són d'entrada i quins de sortida i s'ha configurat la comunicació sèrie amb el mòdul Bluetooth.

#### **Loop**

Un cop el microcontrolador ha finalitzat el setup, entra a la funció loop la qual es va executant mentre el microcontrolador estigui actiu. Degut a que es pretén que sempre que no s'estigui processant una comanda, el microcontrolador estigui dormint, l'únic que es fa en aquesta funció és cridar a sleep.

#### **Sleep**

En aquesta funció, es prepara al microcontrolador per entrar en mode sleep i així tenir un menor consum. Primer de tot es talla l'alimentació del mòdul bluetooth durant uns instants per forçar que perdi la connexió en cas que estigui emparellat amb algun dispositiu i a més que es torni a posar en mode sleep. Tot seguit habilita una interrupció al pin 2 així quan es rebi una dada per bluetooth, com el pin rx del mòdul

BLE està connectat a aquest pin saltarà una interrupció i es despertarà. Després d'habilitar la interrupció, es posa el microcontrolador a dormir. Quan salta la interrupció, el programa reprèn l'execució i el primer que es fa és inhabilitar les interrupcions donat que ja no es necessiten. Finalment es crida a la funció `get commands`.

### Getcommands

La finalitat d'aquesta funció és consumir les comandes que es reben per comunicació sèrie des del mòdul bluetooth. Per evitar que els dispositius hagin d'estar connectats contínuament amb el microcontrolador, es donarà de marge 10 segons entre comanda i comanda. Si en aquest temps no es rep cap comanda nova, es sortirà de la funció i tornarà al loop.

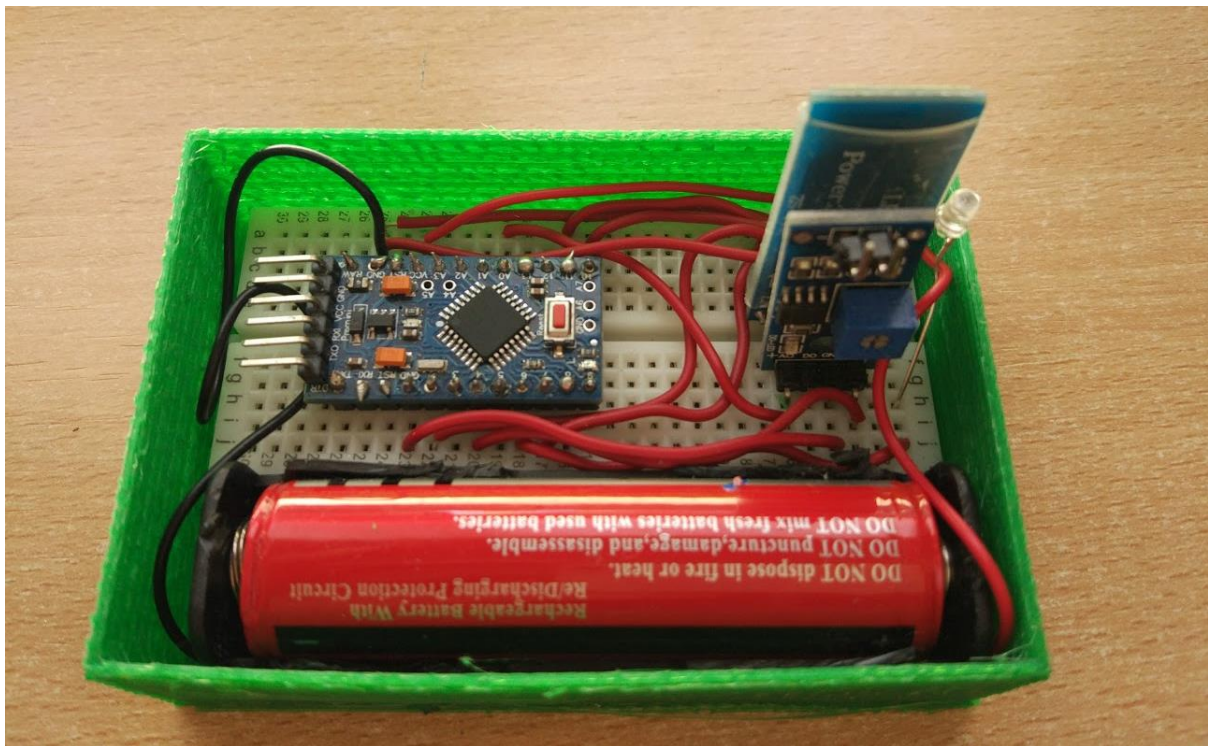
S'han definit 2 classes de comandes: per sol·licitar informació i per interacció. Totes les respostes segueixen el mateix format, en cas que la comanda sigui vàlida, es retorna la comanda + la resposta (si hi ha). Per contra, si es sol·licita una acció incorrecte, es retorna ERROR. En la següent taula es pot veure les comandes disponibles per un dispositiu que tingui un sensor d'humitat i un llum.

Comanda	Resposta Correcte	Descripció
SDEV	SDEV DISPOSITIU 2	Nom del dispositiu i nombre de subdispositius disponibles
SDEV1	SDEV1 11	Nom i tipus del subdispositiu, per exemple 11 = llum
SDEV2	SDEV2 HUMIDITY 1	Nom del subdispositiu 1
ON LED	ON LED	Encendre l'actuador
OFF LED	OFF LED	Apagar l'actuador
VAL HUMIDITY	VAL HUMIDITY 545	Valor del sensor d'humitat



### 6.1.3. Muntatge

Per tal que el circuit es pogués modificar de forma ràpida, s'ha decidit muntar-lo en una protoboard. Aquestes plaques permeten fer connexions ràpides entre components electrònics sense necessitat de soldar. Per protegir el circuit, s'ha dissenyat una caixa així com una tapa i ambdós s'han construït fent ús d'una impressora 3D. El resultat es pot apreciar en la següent figura:



*Figura 6.4 Prototip funcional*

## 6.2. Exemple d'ús: Rec automàtic amb notificació

Com a conclusió del treball, s'ha decidit utilitzar el prototip construït juntament amb el software desenvolupat per controlar el rec d'una planta. El funcionament del sistema serà el següent: s'afegirà a Home Assistant un sensor d'humitat i una bomba d'aigua utilitzant l'aplicació desenvolupada. Quan HA detecti que la terra està massa seca, activarà la bomba d'aigua i a més enviarà una notificació per telegram.

Per implementar aquesta funcionalitat, s'ha modificat el prototip eliminant-ne el led i afegint dos elements nous: Un mosfet i una bomba d'aigua.

Degut a que la bomba d'aigua té un consum major al que pot oferir la sortida d'un pin d'Arduino, no es podrà connectar directament on hi havia el Led. Per solucionar aquest problema se l'alimentarà directament des d'una bateria i es controlarà la seva activació des d'un *mosfet*.

Un *mosfet* és un transistor amb tres entrades anomenades "gate", "drain" i "source" que es pot utilitzar com a amplificador o com a interruptor. Per utilitzar-lo com a interruptor, cal controlar el voltatge entre les entrades gate i source. En cas que no hi

hagi un diferencial de voltatge entre aquests dos, hi ha una resistència elevada entre drain i source cosa que impedeix el pas de la corrent. En canvi si s'aplica una diferència de potencial entre gate i source, la resistència entre drain i source es redueix i permet el pas de corrent. El mosfet escollit és un Irf520 i està muntat en una placa per facilitar la seva connexió.

Per tal d'augmentar l'autonomia del circuit, s'ha afegit una segona bateria 18650 que alimenta exclusivament la bomba.

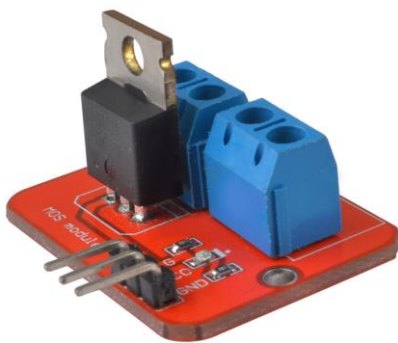


Figura 6.5 Mosfet driver Irf520

Amb el circuit muntat, es procedeix a prendre'n el control amb Network Manager i s'envia a Home Assistant. Un cop fet, des del panell de HA es crea la regla d'automatització per controlar el rec. Com a trigger s'utilitza el valor del sensor d'humitat i les accions seran activar la bomba d'aigua i enviar una notificació per telegram.



*Figura 6.6 Rec automàtic amb notificació*

## 7. Planificació

El projecte s'ha dividit en quatre fases, a continuació es descriu cadascuna d'elles així com la seva duració estimada.

1. **Estudi d'alternatives (50 h):** L'objectiu principal de la fase és seleccionar les tecnologies que es faran servir en el projecte.
2. **Disseny (100 h):** En aquesta fase s'analitzarà les arquitectures proposades en la fase d'investigació, es veurà si són factibles i es seleccionarà la que es consideri més adequada per desenvolupar-la.
3. **Desenvolupament (550 h):** Un cop escollida l'arquitectura per al projecte, es passarà a implementar-la. En aquesta fase es poden distingir els següents agents del sistema:
  - a. Network Manager **(400 h)**
  - b. Home Nexus **(150 h)**
4. **Prototip (100 h):** Construcció de diferents prototips per tal de comprovar el bon funcionament del sistema.
5. **Memòria (50 h):** Redacció de la documentació del projecte. La memòria s'ha escrit en paral·lel al desenvolupament del projecte.

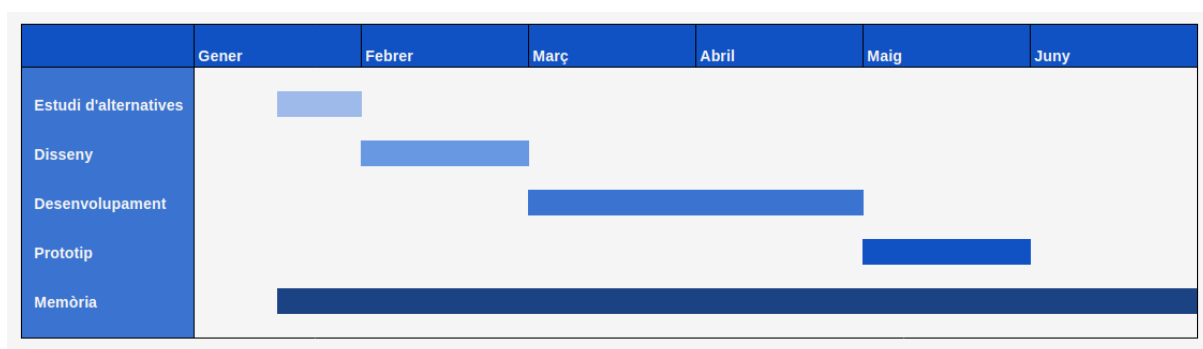


Figura 7.1 Planificació del projecte

## 8. Pressupost

Alhora d'analitzar el pressupost del projecte, es poden distingir tres categories: Costos de Hardware del prototip, costos de material personals. Degut a que tot el programari que s'ha utilitzat és lliure, no hi ha costos de software.

### 8.1. Hardware dels prototips

Material	Preu unitari (€)	Unitats	Cost Total (€)
Placa basada en Arduino Pro mini	1,36	2	2,72
Mòdul bluetooth AT-09	1,77	2	3,54
Protoboard	1,85	2	2,70
Bateria 18650	2,50	3	7,50
Led	0,10	2	0,20
Sensor Humitat	0,83	2	1,66
Irf520 mosfet driver	2,93	1	2,93
Bomba d'aigua	1,10	1	1,10
Capsa impresa amb 3D	1,05	2	2,10
<b>Total</b>			<b>24,45</b>

### 8.2. Personal

La part amb més cost del projecte recau en el salari de les persones que el duen a terme. S'ha diferenciat dos tipus de treballadors: L'enginyer de sistemes, que suposaria un cost de 45 euros / hora i un programador amb un cost de 32 euros / hora.

Tasca	Hores necessàries (h)	Preu Hora (€/h)	Cost total (€)
Estudi d'alternatives	50	45	2.250

Disseny	100	45	4.500
Implementació: Network Manager	400	32	12.800
Implementació: Home Nexus	150	32	4.800
Prototips	150	32	4.800
Redacció de la memòria	50	32	1.600
<b>Total</b>	<b>900</b>		<b>30.750</b>

### 8.3. Equipament

A més del Hardware utilitzat en el prototip i el cost del personal, cal afegir els següents costos relacionats amb l'equipament necessari a l'hora de desenvolupar el projecte:

1. **Ordinador:** Per desenvolupar el software cal un ordinador. Si el cost d'aquest és d'uns 700 euros, la seva vida mitjana és d'uns 5 anys i s'ha fet servir 6 mesos, el cost de l'ordinador seria de 70 euros.
2. **Oficina:** Serà necessari un local on dur a terme el desenvolupament. Com que no cal gaire gran, es preveu un preu d'uns 1000 e/mes. En els 6 mesos que dura el projecte, el cost total serà de 6000 euros.

### 8.4. Cost total

El cost total del projecte resultaria de sumar els costos mostrats prèviament.

Tipus de cost	Cost Total (€)
Prototips	24,45
Personal	30.750
Equipament	6.070
<b>Total</b>	<b>36.844,45</b>

## 9. Conclusions

Utilitzant els prototips construïts, s'ha comprovat que el sistema dissenyat per crear la xarxa de dispositius funciona correctament. Amb el procediment dissenyat, l'usuari pot prendre el control d'un dispositiu de forma senzilla i ràpida.

El Bluetooth Low Energy (BLE) ha resultat ser una bona elecció: amb aquesta tecnologia els dispositius es poden controlar directament des d'un smartphone o un ordinador. A més com que el BLE té un consum molt reduït, els dispositius poden aguantar mesos sense necessitat de recarregar-se. Per realitzar una connexió Bluetooth, normalment s'escaneja en cerca de dispositius i es selecciona el desitjat en una llista. Aquest procés es pot evitar si es coneix la direcció mac del dispositiu en qüestió. S'ha decidit obtenir aquesta informació emmagatzemant-la prèviament en un tag NFC.

La creació de la xarxa es fa utilitzant l'aplicació per a smartphone desenvolupada. Per detectar un dispositiu l'usuari solament ha d'apropar-li el smartphone. Al llegir el tag NFC, s'iniciarà automàticament la connexió i es reconeixerà els diferents sensors o actuadors que té el dispositiu. Després d'emmagatzemar aquesta informació, l'usuari pot controlar l'actuador o llegir el valor del sensor en qualsevol moment solament prement un botó.

En un món on Internet of things està en constant creixement sobretot a la llar, la interacció entre dispositius per automatitzar processos aporta molt valor. Gràcies a la integració de la plataforma d'automatització per a smarthomes Home Assistant amb la xarxa de dispositius, es permet a l'usuari crear interaccions entre els diferents sensors i actuadors que ha afegit a la xarxa i crear per exemple una llum que s'encengui amb un detector de proximitat. A més es dona més llibertat a l'usuari, ja que els sensors són escanejats periòdicament i es poden controlar tots els dispositius des de qualsevol aparell connectat a la mateixa xarxa WiFi o fins i tot, des de qualsevol part del món si es decideix obrir el port en el qual treballa Home Assistant.



## 10. Bibliografia

En aquest apartat es poden veure els enllaços a les fonts d'informació utilitzades en el desenvolupament del projecte.

### Tecnologies de comunicació:

- <https://blog.securifi.com/zigbee-vs-z-wave-whats-the-difference>
- <https://inovelli.com/z-wave-vs-zigbee-vs-bluetooth-vs-wifi-smart-home-technology/>
- <https://www.link-labs.com/blog/bluetooth-vs-bluetooth-low-energy>
- <https://www.digitaltrends.com/mobile/nfc-explained/>
- <http://time.com/3834259/wifi-how-works/>
- <https://elandroidelibre.lespanol.com/2015/08/todo-sobre-zigbee-la-tecnologia-ultrabarata-para-comunicacion-inalambrica.html>

### Plataformes d'automatització:

- <https://randomnerdtutorials.com/9-home-automation-open-source-platforms-for-your-projects/>
- <http://codepop.com/open-sourcecraft/episodes/paulus-schoutsen/>
- <https://home-assistant.io/>
- <https://www.openhab.org/>
- <https://opensource.com/life/16/3/5-open-source-home-automation-tools>
- [https://www.reddit.com/r/homeautomation/comments/6abcr0/home\\_assistant\\_vs\\_openhab\\_2/](https://www.reddit.com/r/homeautomation/comments/6abcr0/home_assistant_vs_openhab_2/)
- <https://seanb.co.uk/2017/03/home-assistant-review/>
- <https://ricveal.com/blog/home-assistant/>

### Network Manager:

- <https://developer.android.com/studio/index.html>
- <https://developer.android.com/develop/index.html>
- <http://www.wikihow.com/Execute-HTTP-POST-Requests-in-Android>
- <https://code.tutsplus.com/tutorials/reading-nfc-tags-with-android--mobile-17278>



- <https://www.allaboutcircuits.com/projects/how-to-communicate-with-a-custom-ble-using-an-android-app/>

### Home Nexus:

- <https://home-assistant.io/docs/>
- <https://github.com/lanHarvey/bluepy><https://github.com/lanHarvey/bluepy>
- <https://tinyio.wordpress.com/2016/04/05/communication-between-raspberry-pi-and-multiple-arduinos-via-bluetooth-low-power-ble/>
- <https://pypi.python.org/pypi/paho-mqtt/1.2>
- <http://docs.python-requests.org/en/master/>

### Prototips:

- <https://www.arduino.cc/en/Main/Software>
- <https://www.arduino.cc/en/Main/Documentation>
- <http://www.instructables.com/id/Arduino-AND-Bluetooth-HC-05-Connecting-easily/>
- <https://evothings.com/control-an-led-using-hm-10-ble-module-an-arduino-and-a-mobile-app/>